

COMPUSA

FDC-100 SERIES

USERS MANUAL

COMPUSA CORPORATION
1101 Bristol Road
Mountainside, New Jersey 07092 U.S.A.
(201) 654-7220 Telex: 138275

Short List of Commands (see each section for details) :

COMMAND -----	FORMAT -----
Save BASIC Program	SAVE"A:FILENAME,P"
Load BASIC Program	LOAD"A:FILENAME,P"
Delete File	SAVE"A:FILENAME,D"
Save Data New File	SAVE"A:FILENAME,V:A\$,N"
Load Data File Track n	LOAD"A:FILENAME,V:A\$,T:n"
Format Disc	SAVE"A,FMT,T:40,DD"
Get Directory	LOAD"A,DIR"

List of all errors generated by FDC :

- G - the disc is full; all the tracks have been used
- H - the disc or file is write protected
- I - the directory of the disc is full
- J - the specified file does not exist
- K - the program is too big to be saved; there is no memory left for buffers
- L - there was a serious, non-recoverable, error during reading or writing the track
- M - there is an unrecognizable parameter or the file or variable name has been specified incorrectly or the variable does not exist
- N - the file already exists; the program has not been saved
- P - an illegal combination of parameters has been given
- R - the track number is out of range
- S - there was an error in reading or writing the directory

In addition the following are the errors most frequently generated by the Sinclair ROM :

- O - Successful completion.
- 4 - Not enough room in memory
- D - Program interrupted by BEAK or STOP

Contents

	page
1.0 Introduction	1
2.0 The Hardware	3
2.1 Floppy Discs and Disc Drives	
2.1.1 Density and Sides	
2.1.2 Write Protect	
2.1.3 Tracks	
2.1.4 Soft Sectorred Discs	
2.1.5 Drive Select	
2.1.6 Types of Drives	
2.2 Installing the Disc Drives and FDC	
2.3 Using the Disc Drives	
2.4 Start Up	
2.5 The Power Supply	
3.0 The Filing System	10
3.1 The Directory	
3.1.1 The Directories in RAM	
3.1.2 Files	
3.1.3 File Status	
3.2 Formatting	
3.3 Bad Tracks and Errors	
3.4 File Handling	
3.4.1 Programs on Disc	
3.4.2 Data File Input and Output	
3.4.3 Direct Input and Output	
4.0 The Disc Commands.....	18
4.1 The Load Program Command	
4.2 The Save Program Command	
4.3 The Read File Track Command	
4.4 The Write File Track Command	
4.5 The Delete File Command	
4.6 The Change File Command	
4.7 The Format Disc Command	
4.8 The Get Directory Command	
4.9 The Read Physical Track Command	
4.10 The Write Physical Track Command	
5.0 Error Reports	30
6.0 Machine Code	31

(C) COMPUSA CORPORATION 1983, all rights reserved. No part of this document may be copied without prior written consent from COMPUSA.

Issue 1.0 12/7/83

FDC-100 SERIES MANUAL

1.0 Introduction

The FDC-101/102 are two floppy disc systems for the Sinclair ZX-81, Timex 1000 and 1500. They are single and dual drives systems respectively, complete with FDC-100 drive controller, one or two disc drives, one dual drive case, drive power supply and all cables and connectors.

The FDC-100 is a floppy disc controller which is capable of controlling one or two single or double sided, single or double density drives. It will work with all Shugart compatible 5 1/4" disc drives.

There are a number of 'Shugart Compatible' 3", 3-1/4" and 3-1/2" miniature drives which are becoming available. These should work with the FDC, but as there are no standards for the specification, we cannot guarantee it. This manual gives you enough information to decide which non-standard drives will work.

The controller will not work with other disc drives, such as

- "Apple" drives
- Micropolis (or compatible) drives
- 8" drives

The FDC includes a Disc Operating System in ROM (Read Only Memory). This gives a simple disc interface to BASIC involving no USR calls, PEEK, POKE, etc. The DOS commands include,

- Load or Save a program
- Read or Write a data file record
- Delete a file
- Read disc directory
- Write/delete protect a file
- Format a disc
- Automatically load and execute a program on power-up

All these commands are available directly from a BASIC program or from the BASIC command mode using LOAD and SAVE commands. The computer can still LOAD and SAVE to cassette. The computer bus is continued to allow connection of RAM packs and other peripherals.

No modifications are necessary to the computer when fitting the FDC. However, you will need extra RAM - we recommend at least 16K but the controller will work with up to 64K extra. Read the section on Installing the FDC to check whether your RAM pack and peripherals will be O.K.

You will not be able to use the FDC to read discs written by another make of computer or controller. The FDC writes the data in a special format and also, the computer stores its BASIC programs in a non-standard way. (In fact, the above is not quite true. The FDC should be able to read and write any FM or MFM coded floppy disc in any format so long as it does not have missing clock pulses but you will have to do a lot of work in machine code. If

FDC-100 SERIES MANUAL

you succeed, send a copy of your program to the FDC User's Group c/o COMPUSA for distribution to others.

The FDC User's Group will distribute Public Domain software donated by its members. The clearing and distribution of the software will be done by COMPUSA at a very low cost. Copies of the User's Group discs will be available in two ways 1) submit a program to the library and receive a copy of the disc on which your program appears. 2) pay the disc cost plus postage and handling. Please call COMPUSA (201-654-7220) for the current charge.

This is the a formal specification of the FDC:

- Density: single or double
- Coding: FM or MFM
- Data rate: FM 125k bits/S
MFM 250k bits/S
- Precompensation: 500nS, early and late on all tracks in MFM
- Power requirements: 5VDC, 0.31A
- max. number drives: 2
- max. number sides: 2
- max. tracks/side: 124
- sectors per track: 1
- assumed head load time: 50 mS
- assumed track step time: 35 mS
- assumed motor start time: 3 revs
- track format:
 - lead in (55h): 30 bytes
 - sync (FEh): 2 bytes
 - track number: 1 byte
 - valid data length: 2 bytes
 - data: 2900/5800 bytes
 - checksum: 1 bytes
 - sync (0FEh): 2 bytes
 - trailer: 2 bytes
- write retries: 6
- read retries: 6

2.0 The Hardware

This section includes information of interest to users of the FDC-100 who have purchased their own drives and users of the FDC-101/102 who have purchased the complete system from COMPUSA.

2.1 Floppy Discs and Disc Drives

A floppy disc consists of a film of plastic with a magnetic coating on both sides. This is enclosed in an envelope to protect the surfaces from damage.

2.1.1 Density and Sides

The FDC-100 can be used with disc drives which can write to (record on) either one or both sides of the disc. The pattern recorded can be either in "single density" or "double density". Double density recording requires better quality equipment but records twice as much data on each side of the disc.

To be able to write in double density, you will need a double density drive. However, a double density drive can also record in single density. Similarly, to write on both sides of the disc, you will need a double sided drive.

Note - any information written to the "second" side of a double sided disc by a double sided drive CANNOT be read by a single sided drive. The second side of the disc turns in the wrong direction. This is not the same as cutting a write protect notch on the back of a disc, turning it upside down and writing to it with a one sided drive. We will leave the debate on the advisability of this to others. We never do this with any important data.

The FDC-101/102 is supplied with singled sided, double density, 40 track drives. It is possible to mix types of drives on the same system, but be wary of writing to one drive and not being able to read it with the other.

The discs themselves can be either double or single density and double or single sided. A single sided, single density disc might be capable of recording on both sides at double density - but the manufacturer has not tested it for this and does not recommend it.

You will see 5-1/4" discs advertised as DS/DD, SS/SD, etc. SS and DS mean single and double sided; SD and DD mean single and double density.

2.1.2 Write Protect

The envelope of a 5-1/4" disc has a notch cut in the side. If the notch is not there, a switch in the drive stops the drive from recording on the disc - the disc is said to be "Write Protected". If you try to write to the disc, your BASIC program will stop with

an error report.

2.1.3 Tracks

The drive writes to and reads from the disc with a read/write "head". This can be positioned at various distances from the center of the disc and so records concentric circles of data called "tracks".

The number of tracks written on one side of a disc is different for different types of drive. A 5-1/4" disc recorded on a 96 tracks per inch (t.p.i.) drive cannot be read on a 48 t.p.i. drive and vice versa. A 48 t.p.i. drive will record 40 tracks on one side of a disc.

You will see from this that if you want to swap 5-1/4" discs with your friends, or buy software on disc, you will have to specify exactly what kind of drive you have. All of this information can be obtained by asking the dealer who sold you your drives.

2.1.4 Soft Sector Discs

There is a small hole in a 5-1/4" disc called the index hole. The drive can sense the index hole as the disc rotates and uses it to tell where to start reading or writing. Hard Sectored - discs have several holes and will not work with the FDC. You must only use "Soft Sectored" discs.

2.1.5 Drive Select

The FDC-101 has one drive jumpered as "A" and the FDC-102 has two drives jumpered as "A" and "B". The FDC will only support two drives and may produce strange or no results with a drive jumpered as other than "A" or "B".

The FDC sends exactly the same signal to both drives all the time, however, when you select "Drive A" or "Drive B", only one of the drives responds. Each drive has a set of switches or wire links which tell it what number it should respond to. 5-1/4" drives generally respond to one number out of four.

If the drive select options are numbered 0, 1, 2 and 3, then the FDC will select drive 0 as "Drive A" and drive 1 as "Drive B".

The select lines of some makes of drive are numbered 1, 2, 3 and 4, in which case, the FDC will select drive 1 as "Drive A" and drive 2 as "Drive B".

If you have only one drive, it is usual to use it as "Drive A". When you buy your drive, it will have the switches, links or whatever set so that the drive responds either to every number or as drive 0 (or 1). This should be O.K. if you only have a single drive, but must be changed if you have a two drive system. Ask your dealer how to change the drive select option - you may have to solder some wires, cut tracks on the printed circuit board or

flick switches.

If you have only one drive which responds as both "Drive A" and "Drive B", you may get odd results if you write a file to "Drive A" and then try to read it back from "Drive B".

2.1.6 Types of Drive

As discussed above, 5-1/4" discs can be either double or single density and double or single sided. 5-1/4" drives are available for double and single sided, double or single density discs and with various numbers of tracks per side. There is not yet a standard for the 'Shugart compatible' 3" to 3.5" drives but there will probably turn out to be a similar range of options.

2.2.1 Setting up the FDC-101/102

The system comes complete with one  two drives and power supply.

- Remove the six small screws holding the drive case together.
- Feed the power and drive cable through the rectangular opening in the case.
- Plug the drive cables in as indicated on the connector.
- The power connector is keyed and only plugs in one way.
- Remove the shipping protector(s) from the drive(s).

2.2.2 Using the FDC with other Disc Drives

When you buy your 5-1/4" disc drive, it may contain a "shipping insert" - a piece of cardboard shaped like a disc. This protects the head during transit but should be removed before you try to put in one of your discs.

The 5-1/4" flat "ribbon" cable coming out of the FDC must be plugged into the back of the drive with pin 1 of the cable connected to pin 1 of the drive. The cable comes out of the FDC with pin 1 furthest from you as you type at the computer. It should be marked with either:

- a colored band on the cable (plain cables)
- a brown cable ("rainbow" cables)
- an arrow on the connector

You should ask your dealer which is pin 1 of your 5-1/4" drive; there should be a slot in the board (near the pin 1 end) and a key in the connector to prevent you getting it wrong.

Do not plug in or unplug the ribbon cable with power connected to the drive or the FDC.

In order to prevent noise on the drive control lines, the drive

manufacturers provide optional pullup resistors. If you have several drives on a single cable only the last one of them needs to have the pullups connected. This means that all drives should have some way of removing the pullups. Some drives have 150ohm pullups but our controller will not work with these. We recommend the use of 470 to 1.5k ohm resistors. The dealer who sold you the drive will be able to tell you how to unplug the pullup resistor pack and plug in a 470 to 1500 ohm one - they cost about \$1.00. (In fact, all the drives we've tested so far have worked without any pullups at all but we hesitate to advise this. You will not damage a drive by operating it without pullup resistors, so try it and see.)

You must not obstruct the flow of air around the disc drive or the FDC or they may overheat.

2.2.3 Memory and Port Requirements

We recommend at least 16K extra RAM but the controller will work with a 64K RAM pack. (It should work with only 8K of RAM but we don't have an 8K pack to try it with.) The controller's software resides in a ROM at address 3000h to 3FFFh (12268 to 16383). If you are using a 64K RAM pack, make sure that it does not supply memory between addresses 3000h to 3FFF (12268 to 16383). None of the 64K packs that are available supply a full 64K of RAM (or they would clash with the Sinclair/Time BASIC ROM):

- some supply 48K from 4000h to FFFFh (16384 to 65535)
- some supply 58K from 2000h to FFFFh (8192 to 65535)
- and some have switches to switch in or out the section from 2000h to 3FFFh (8192 to 16383).

Also, check that your peripherals (joysticks, sound effects, etc.) don't use "Memory mapped I/O" in this region.

The FDC uses I/O ports #01, #03, #41 and #43. If you have some non-standard peripherals attached, check that they do not use the same ports. Also, none of your peripherals must interrupt the Z80 or affect the WAIT line during disc operations.

2.3 Using the Disc Drives

To insert a 5-1/4" disc:

- open the door by lifting the flap or pressing the button
- insert the disc, being careful not to bend it, with the label facing up.

To remove the disc:

- open the door by lifting the flap and take out the disk or
- push the button and the disc will pop out.

If the disc is Write Protected, the drive can only read from it, it cannot record. To write protect a 5-1/4" disc, fold a self-

FDC-100 SERIES MANUAL

adhesive label over the notch in the side of the disc. You should get a few small tabs with each box of discs. Otherwise, a small piece of sticky tape is usually O.K. but:

- some drives use a light beam and will "see through" the tape: tape a small piece of paper over then notch
- some drives use switches which press hard on the notch and will squash the tape
- make sure the tape won't fall off inside the drive - the envelopes seem to be specially non-stick
- don't have a sticky surface exposed to carry dust into the drive.

Floppy discs are reasonably robust but you should take care in handling and storing them:

- do not touch the recording surfaces
- return the disc to the jacket after use
- do not write on the label with heavy pressure
- do not bend them
- do not store them near magnetic fields
- store them at room temperature (10C - 50C)

The only maintenance required for drives is occasional cleaning of the heads. Follow the manufacturers instructions or purchase a commercially available head cleaning kit. How often you will need to clean the disc drives depend on the quality of the drive, of the discs you use and how much you use them. Usually, when the heads are dirty, the drive will read alright but write badly.

2.4 Startup

Firstly, connect the controller into the back of your computer with the RAM pack (but without the disc drive) and switch on. A message saying:

```
0 REM COMPUSA FDC-100 V.n
```

should appear at the top of the TV screen (n is the version number and is .5 as of this date).

If it doesn't work, check that you've got good connections. If you are using a 64K RAM pack, try the FDC with a borrowed 16K pack.

Now switch off and try it with a disc drive plugged in but without a disc in. Make sure the drive is plugged in the right way around. The controller should fail to read anything from the drive and again come up with the "COMPUSA DISC SYSTEM" message. If it doesn't, check the pullup resistors and the drive select options on the drive.

Now put a disc in the drive and use the Format Disc command to format the disc. The drive motor should start and you should hear a click as the drive head is loaded. Assume at first that your

FDC-100 SERIES MANUAL

drive is single density, single sided, with only a small number of tracks:

```
SAVE "FMT,A,T:20"
```

Check that you can write and read a file with:

```
LET S$="ABCDEF"
SAVE "A:FILE,N,V:S$"
LOAD "A:FILE,T:1,V:T$"
PRINT T$
```

This should print "ABCDEF".

Now try formatting the disc properly with the right density, number of tracks and sides. The command for the FDC-101/102 is:

```
SAVE "A,FMT,T:40,DD"
```

2.5 The Power Supply

The FDC can take power from one of several sources:

- the 5V line on the computer bus
- the 9V line on the computer bus (with optional regulator)
- an external 5V
- an external 9V (with optional regulator)

The computer regulator should be able to supply enough current at 5V for the computer itself, the FDC and a RAM pack. But if you have more plugged in, it may not be able to cope.

The computer takes its supply from the 9V adapter. This 9V is also connected to the computer bus. The FDC can drop this to 5V for its own use, but passes on to the next object on the bus (e.g. a RAM pack) the 5V regulated by the computer. The 9V power supply requirements of a typical system are:

T/S computer	0.34A
Sinclair 16K RAM	0.21A
COMPUSA FDC-100	0.31A

	0.84A

So a 0.7A Timex or Sinclair adapter may not cope - some do, some don't. If you overload the mains adapter, the first thing you see is a kind of "bulge" moving up or down the TV screen. (We are told that the original ZX81s were supplied with a 0.7A adapter, but to drive a printer, you then had to buy a 1.2A adapter - We think the later Timex 1000s and ZX81s were supplied with a 1.2A adapter as standard.)

With a printer and other peripherals taking current from the 9V as well, even a 1.2A adapter may not cope. In that case, you will either have to buy a better adapter, or give the FDC it's own 9V supply from a second adapter. The regulator in the computer is

FDC-100 SERIES MANUAL

able to supply 1A at 5V. At 1A however, the heat sink gets quite warm - it might be best to fit a bigger heat sink. You shouldn't be able to damage the regulator by taking too much current: it limits the current it supplies and also turns off if it overheats.

We have had success using existing Timex adapters by adding a large electrolytic capacitor (1000 uf or greater) across the 9V of the adapter.

Finally, there are two connection points on the FDC (for +5V and ground). The FDC can be run from an external 5V power supply; in fact, the whole computer and accessories could run off this 5V (but check that your RAM pack doesn't need the 9V line). 5-1/4" drives usually have an internal +5V and +12V supply, you might like to use this +5V to power your computer, but check that it can supply enough current at 5V.

There are various links on the FDC printed circuit board which can be cut (with a sharp knife) to set up these various options. Figure 1 shows where they are and how they're connected. They should be cut as follows:

- FDC takes its 5V from the computer. This is how the FDC101/102 is shipped.
- FDC takes 9V from computer bus: install the FDC regulator and smoothing capacitor. Cut link LK1.
- FDC takes 9V from a separate adaptor: install the FDC regulator jack socket and smoothing capacitor. Cut links LK1 and LK2. When the jack plug is not inserted, the FDC will take 9V from the computer bus.
- FDC takes 5V from an external 5V power supply: do not install the FDC regulator, jack socket or smoothing capacitors. Cut link LK1.
- FDC and computer system take 5V from an external 5V power supply: do not install the FDC regulator, jack socket or smoothing capacitors. Cut no links. Do not plug an adapter into the computer.

The FDC-101/102 is supplied with wall mount adapters which provide +5VDC and +12VDC with sufficient current capacity to use the drives supplied. If you are using the FDC-100 with your own drive you will have to provide a power supply for them.

FDC-100 SERIES MANUAL

3.0 The Filing System

3.1 The Directory

In the previous section, we described what a disc looks like to the FDC but to a BASIC program it looks very different.

When you save data or a program on a disc you give it a name; then, when you want to read it back, you ask for it by that name. A named set of data or a named program stored on disc in this way is called a "File".

The way that the space on the disc is divided into files is specified by the disc "Directory". The directory also holds a description of the disc:

- single or double sided
- single or double density
- the number of tracks on each side

The directory is always held on track zero (the outermost track) of side zero of the disc and is always in single density. The FDC can therefore always read the directory without knowing anything about the drive.

3.1.1 The Directories in RAM

When the computer is switched on, the FDC reads the directories of the discs in both drives. If it succeeds, it will keep a copy of these in its memory (RAM).

To save time, when reading data from a file, the FDC will not read the directory but will look at the copy in RAM to see where the file is stored on the disc. But the FDC cannot tell if you have taken the disc out and put in another one, so it's possible it could read the 'right' place on the 'wrong' disc. However, writing data to the wrong disc could be disastrous so the FDC always reads the directory before any alteration.

In addition, the Get Directory command always reads the specified disc. So you should do a Get Directory after changing the disc and at the start of any program which reads disc files.

If your BASIC program goes wrong, it might alter the RAM copies of the directories. So a "checksum" is kept with them. If the FDC finds this checksum does not match the data in the RAM directories, the directories are read from the discs.

The directory is always written to the disc whenever it is changed. If the copy of the directory on the disc becomes damaged, perhaps because the disc is worn, the FDC will not know what tracks belong to which files and you will be unable to read the files later.

However, so long as you have a good copy of the directory in RAM, you will be able to read the files (but not write to them). If you ever get an error in reading or writing a directory, immediately copy the programs to a different disc in the other drive. You will then have to format the bad disc.

BECAUSE YOU CAN LOSE ALL THE FILES ON A DISC IN THIS WAY OR IF THE DISC IS DESTROYED, YOU SHOULD ALWAYS BACKUP ALL IMPORTANT FILES ON A DIFFERENT DISC. (No-one ever follows this advice until it happens to them!)

The RAM directories are kept in the computer memory above (RAMTOP). If you want to write machine code programs in this area, you should read the section on Machine Code.

3.1.2 Files

As mentioned above, the data on the disc is divided into "Files". A file can contain one BASIC program or data such as a string or an array of numbers.

You can create a file by saving the current BASIC program in the computer memory or by writing data to the file. The file commands are discussed below, but, for instance,

```
SAVE "A:STARTREK,P"
```

will save the current program on disc A and give it the name STARTREK.

Every file on a disc must have a different name, though two files with the same name may exist on different discs. The name consists of up to 10 characters but should not include a comma or any character above 127 (see the computer manual). In general, it's best not to use characters above "Z".

A file can be of any size up to the total size of the disc (minus the track for the directory of course). A file is written to zero or more tracks. It can fill a part of a track but then no other file can use the rest of that track.

A new disc will initially contain no files, so all the tracks are free. As you store files on the disc, this "Pool" of "Free Tracks" gets used up until the disc is full.

However, you can delete files from the disc and return the tracks they were using to the Free Pool. For instance,

```
SAVE "A:STARTREK,D"
```

will delete the STARTREK program we saved before. The file will disappear from the directory and now cannot be read.

The tracks chosen from the Free Pool are 'rotated' so that no part

of the disc data are gets worn more than any other. Unfortunately, the directory must always be on track zero so it tends to get worn a lot.

3.1.3 File Status

If you want to protect a file from being deleted or changed, you can "Write Protect" it. For instance,

```
SAVE "A:STARTREK,WP"
```

will write protect our STARTREK file. It can't be altered or deleted until we "Write Enable" it with:

```
SAVE "A:STARTREK,WE"
```

Write protecting a file is different from write protecting the whole disc with the write protect tab (see above). To be able to delete or alter a file, both it and the disc must be write enabled.

If you PRINT the directory of a disc, you will see what files it contains and their statuses:

```
LOAD "A,DIR,V:S$"
PRINT S$
```

or just

```
LOAD "A,DIR"
```

This will print something like:

```
,FREE DD      5
STOCK          P 5
STARTREK       P 3
ACCOUNTS WP   P 2
MAP            14
MENU           WP S P 1
```

The ,FREE entry tells you how much free area is left on the disc (the name contains a comma so you can't create a file of the same name).

The other entries show you what files are on the disc. The number after each name tells you the size of the file in tracks and the remaining letters tell you the status of the file.

A "WP" means that the file is Write Protected; a "P" means that the file was saved as a program not as data, and an "S" means that the file is a "Startup File".

The "P" is only there for your convenience. If you ask it to, the FDC will treat a Program file as data or load a Data file as though it is a program.

A Startup file is a special kind of Program file. When the computer is reset (when it is switched on), the FDC reads both directories, then searches them for a Startup file. First it searches the A disc, then the B disc. The first Startup file it finds, it loads into memory. If the file was saved while it was executing (see the computer manual) then it will resume execution.

This feature is very useful if the computer is to be used by people who don't know how to program. You could write a "Startup" program which displays a menu of the games on the disc and asks which one you want to play or which starts your stock control system. The innocent user would then never have to learn to type such uncouth phrases as

LOAD "A:STARTREK,P"

If the FDC encounters an error in the middle of loading a Startup program, it may not be able to recover to give an error message. In which case, it will perform a NEW command with RANTOP set to 5000h (20480). No COMPUSA banner will be generated and no further disc I/O should be attempted. Remove the disc and switch the computer on again.

3.2 Formatting

When you buy a disc, it is completely blank. It has no data on the tracks and more importantly, it has no directory on track zero. So, in order to use the disc, you must write an empty directory onto it. This is called "Formatting" the disc.

This directory will also hold a description of the disc and the drive it is loaded into:

- single or double sided
- single or double density
- the number of tracks on each side (not the t.p.i)

The FDC-101/102 is supplied with one or two singled sided drives which will write in single or double density and have 40 tracks. For other disc drives consult the manual for the drives for these parameters.

In single density, 15625 bytes of data are written per second; in double density, it's 31350 bytes/S. A 5-1/4" 'Shugart Compatible' drive rotates at 300 rpm, so this gives 3125 and 7250 bytes per rotation (track). The drive will not rotate at exactly this speed and cannot start and stop writing immediately, so you must subtract at least 100 bytes from this. If you have a non-standard drive and you try to fit more bytes onto a track than the track will hold, then the end of the track will overwrite the beginning, but too few bytes will never give trouble. In fact, the FDC ROM sets the number of bytes per track to 2900 in single density and 5800 in double density.

If you get any of the parameters wrong then you will either get poor quality recording (with lots of errors) or none at all. If you specify too many tracks per side, then some (cheap and nasty) drives will try to step the head too far and it will get stuck.

The Format command is described in detail below. You can also use the command to delete all the files on the disc (including the Write Protected ones) and return all the Bad Tracks to the Free Pool.

3.3 Bad Tracks and Errors

When a track is written, it is immediately read to make sure that the write was successful (this is why writing takes longer than reading). If the data cannot be read, then the FDC tries again. After several retries, the FDC assumes that the disc has been damaged in that region and selects another track. The directory will always be kept up to date as to which file uses which tracks.

When a bad track is discovered, it is added to the "Pool" of "Bad Tracks". Initially, the disc is assumed to have no bad tracks. You can tell how many Bad Tracks there are on a disc by seeing whether the total tracks in the Free Pool and all the files adds up to the number of tracks on the disc.

If you get a lot of errors, then either:

- the disc heads need cleaning (use a commercially available head cleaning kit)
- you've formatted the disc wrongly (DS or DD on a SS or SD drive or the wrong number of tracks per side)
- there's something wrong with the disc (don't buy the cheapest discs or try to use single density discs in double density)
- there's a bad connection between the FDC and the drive, the computer or the RAM pack
- there's something wrong with the drive (for the FDC-101/102 call COMPUSA for other drives see the dealer that sold you the drives)
- there's something wrong with the FDC (call COMPUSA)

NOTE: DO NOT RETURN ANYTHING TO COMPUSA WITHOUT FIRST CALLING OR WRITING FOR A RETURN AUTHORIZATION NUMBER.

When writing or altering a file, the new version of the file is always written to a new track from the Free Pool. Only when the change is complete, is the old version of the track added to the Free Pool and the file entry in the directory altered to include the new version. In this way, if an error occurs, the 'good' data on the disc corresponds to the directory description up till the last possible moment.

3.4 File Handling

This section discusses File input and output in general.

When writing data or a BASIC program, one or more whole tracks are written. If there is not enough data or program to fill a whole track, then the extra bytes are taken from the computer memory immediately after the data. This extra region must not include anything above where the Z80 stack pointer points (say, 60 bytes less than ERR-SP). If it does, an error 4 will be reported.

When the FDC reads a track, it needs space in the computer memory to put the data. Even if the track holds only a few bytes of valid data, the FDC cannot know this in advance and must request enough space to hold the whole track. If the ZX81 memory is full, an error 4 will be reported.

3.4.1 Programs on Disc

Saving a program on disc is almost exactly the same as a saving a program on tape, except:

- it's faster
- it's more reliable
- you don't get all that mess on the TV
- if the command is in a program, the top bit of the last character of the LOAD or SAVE string does not get set.

If you write the program from a command within the program, it will continue execution when the file is next read. (See the computer manual.)

(The FDC assumes that a "Program" starts at VERSN - 16393 - and ends at the byte pointed to by ELINE - 16404.)

3.4.2 Data File Input and Output

As discussed above, a data file consists of zero or more tracks. A zero length file has an entry in the directory but has no data.

The tracks of a file are numbered from one up to the number of tracks in the file. When you write a data file, you write it one track at a time. You can either add a new tracks to the end or write different data to an existing tracks.

If you rewrite an existing track, the FDC selects a track out of the Free Pool and writes the data to that. Only when the write has succeeded, does it alter the directory so that the file includes this new track.

The data written to the track consists of either:

- a string
- a character array
- a numeric array

These can be of any length up to the track size set when the disc was formatted (double density holds twice as much per track). The lengths of these are:

- string S\$: LEN(S\$)
- character array: Number.of.chars+2*Number.of.dimensions+1
e.g. DIM A\$(25,100) = 25*100+2*2+1 = 2505
- numeric array: 5*Number.of.elements+2*Number.of.dimensions+1
e.g. DIM A(10,2,30) = 5*10*2*30+2*3+1 = 3007

When you read the track back, you should read it into a variable of the same type. If it has been DIMed to a different size, the read will still succeed but the array will be the size and shape it was when the track was written. If the variable does not already exist, the FDC will assume that:

- V:A\$ refers to a simple string variable
- V:A refers to a numeric array

and it will create a variable of that type.

If you read the track into a variable of a different type, you will get odd results but no error message. See the computer manual for details of how variables are stored.

The FDC will not write a track of length zero. If you rewrite zero length data to an existing track, the old track will be removed from the file and the following tracks will be moved down. For instance, if the file DATA has 10 tracks, the code:

```
LET S$="TRACK 7"
SAVE "A:DATA,V:S$,T:7"
LET S$=""
SAVE "A:DATA,V:S$,T:5"
LOAD "A:DATA,V:S$,T:6"
PRINT S$
```

will print "TRACK 7" and the file will have only 9 tracks.

A track of zero length cannot be added to the end of a file. However,

```
LET S$=""
SAVE "A:DATA,V:S$,N"
```

will create a file called DATA if one does not already exist.

3.4.3 Direct Input and Output

When you read or write to a file, you don't know which actual

track of the disc has been selected. If you want to, you can read or write a particular track of the disc - the "Physical" track rather than the "Logical" track of the file. The load or save command simply specifies a disc and a track number but no filename.

The Read and Write Physical commands should be used with caution as they do not use the directory. If you accidentally write over track zero, you will corrupt the directory and you won't know which track belongs to which file. They can be very useful and are provided with this warning.

If a Physical Write fails because the track is Bad, an error will be reported.

4.0 The Disc Commands

All disc operations are performed through BASIC LOAD and SAVE commands. A normal tape LOAD or SAVE command has the following format:

```
LOAD "NAME"
SAVE "NAME"
```

The name of the file is specified in the string enclosed by the quotes. Commands like this will still work with the FDC plugged in. However, certain strings will be interpreted by the FDC as disc commands and no tape input or output will occur. The string part of a FDC command consists of Parameters separated by commas, for instance:

```
SAVE "A:STARTREK,WP,P"
```

This is a SAVE command which means that something is going to be written to the disc. LOAD commands all read something from the disc without changing it. The command string has three parameters

```
A:STARTREK      WP      P
```

The order in which parameters occur is not significant - we could as easily have written:

```
SAVE "P,WP,A:STARTREK"
```

This command means save the current BASIC program (P means program) as a Write Protected file (WP means Write Protected) called STARTREK on disc A. So it is the disc equivalent of a computer SAVE command - except that you can't accidentally delete or write over this particular file.

Of course, things might not go right - the disc or the directory might be full or there might already be a file called STARTREK in the directory of disc A - in which case you will get an error reported in the usual way, e.g.

G/O

means the disc is full. There are a whole new set of error report codes for disc operations. (Refer to the Errors section of each command and Section 5.0 Errors All Commands.)

Each command has its own set of parameters; some are optional and some must be given. If the FDC program can recognize at least two of the parameters, it will assume that the LOAD or SAVE is to disc. Otherwise, it will treat the command as a tape operation and attempt to LOAD or SAVE the current program to or from tape. If any of the parameters is unrecognizable or contains an error, such as too long a filename or a non-existent variable in a SAVE,

then an error M will be reported.

The string argument of the LOAD or SAVE command need not be a constant. For instance:

```
LOAD "A:FILENAME,T:"+STR$(TKNUM)+"",V:A$
```

However, the argument must be evaluated to give a string. The number after a "T:" must be the string representation of a numeric value - not the name of a variable. For instance, if you want to read the track at TKNUM of file FILENAME, you should do it as shown above, not like this:

wrong:

```
LOAD "A:FILENAME,T:TKNUM,V:A$"
```

Similarly, the value after an "A:" or "B:" must be the string representation of the file name not the name of a string variable. For instance, if you want to read the file whose name is in F\$, you should say

```
LOAD "A:"+F$+",T:4,V:A$"
```

and not:

wrong:

```
LOAD "A:F$,T:4,V:A$"
```

Finally, the value after a "V:" must be the name of a variable not the contents of the variable or a constant. For instance, if you want to write the contents of A\$ followed by "END OF FILE", you should say:

```
SAVE "A:FILENAME,N,V:A$"  
LET S$="END OF FILE"  
SAVE "A:FILENAME,N,V:S$"
```

and not:

wrong:

```
SAVE "A:FILENAME,N,V:"+A$
```

wrong:

```
SAVE "A:FILENAME,N,V:""END OF FILE"""
```

4.1 The Load Program Command

The Load program command loads the contents of a file into the BASIC program area of the computer. Usually the file will have been created by saving a BASIC program but the Load Program command may also be used to load a data file as if it were a program.

4.1.1 Format - Load Program

The format of the Load Program command is as follows:

```
LOAD "d:filename,P"
```

Where:

- d is the disc containing the file - "A" or "B"
- filename is the name of the file to be changed
- the P option specifies that it is a program which is to be loaded

4.1.2 Examples - Load Program

```
LOAD "A:STARTREK,P"  
LOAD "P,B:X-3"
```

4.1.3 Errors - Load Program

The following error messages may be generated by the Load Program command:

- J - the specified file does not exist
- L - there was a serious, non-recoverable, error while reading one of the tracks
- M - there is an unrecognizable parameter or the file name has been specified incorrectly
- P - an illegal combination of parameters has been given
- S - there was an error in reading or writing the directory

If the FDC encounters an error in the middle of loading a program, it may not be able to recover to give an error message. In which case, it will perform a NEW command with RAMTOP set to 5000h (20480). No COMPUSA banner will be generated and no further disc I/O should be attempted until the computer has been reset.

4.2 The Save Program Command

The Save program command saves the BASIC program area of the computer into a file. The file will be marked as a Program file in the directory.

While saving the file, you can also mark it as Write Protected or as a Startup file to be loaded when the computer is switched on.

If you save the program from within itself, then when it is loaded it will carry on execution from the line after the SAVE command.

4.2.1 Format - Save Program

The format of the Save Program command is as follows:

```
SAVE "d:filename,P"  
SAVE "d:filename,P,WP"  
SAVE "d:filename,P,S"
```

SAVE "d:filename,P,WP,S"

Where:

- d is the disc containing the file - "A" or "B"
- filename is the name of the file to which the program will be saved the WP option will Write Protect the file
- the S option will make the file a Startup file
- the P option specifies that it is a program which is to be saved

4.2.2 Examples - Save Program

```
-----
SAVE "P,B:X-3"
SAVE "A:STARTREK,WP,P"
SAVE "B:HELLO,P,S"
SAVE "A:ACCOUNTS,P,WP,S"
```

This program will save itself if you type GOTO 30. It will then be loaded and will execute when the computer is switched on:

```
10 PRINT "HELLO"
20 END
30 SAVE "A:HELLO,P,S"
40 GOTO 10
```

4.2.3 Errors - Save Program

The following error messages may be generated by the Save Program command:

- G - the disc is full; all the tracks have been used
- H - the disc or file is write protected
- I - the directory of the disc is full
- K - the program is too big to be saved; there is no memory left for buffers
- M - there is an unrecognizable parameter or the file name has been specified incorrectly
- N - the file already exists; the program has not been saved
- P - an illegal combination of parameters has been given
- S - there was an error in reading or writing the directory

4.3 The Read File Track Command

The Read File Track command reads one track of a file into a variable. The first track of the file is "logical" number one. Subsequent tracks are numbered up from this. An error will be reported if you try to read before the first track or after the last one.

4.3.1 Format - Read File

The format of the Read File Track command is as follows:

LOAD "d:filename,T:tracknum,V:variable"

Where:

- d is the disc containing the file - "A" or "B"
- filename is the name of the file to be changed
- variable is the name of the variable into which the track will be read
- tracknum specifies which track of the file should be read

4.3.2 Examples - Read File

```
-----
LOAD "A:DATA-5,T:7,V:T$"
LOAD "T:1,V:V$,B:CASH"
```

4.3.3 Errors - Read Files

The following error messages may be generated by the Read File Track command:

- 4 - there is not enough room in memory to create a variable to hold the data
- J - the specified file does not exist
- L - there was a serious, non-recoverable, error during reading
- M - there is an unrecognizable parameter or the file or variable name has been specified incorrectly
- P - an illegal combination of parameters has been given
- R - the track number is out of range
- S - there was an error in reading the directory

4.4 The Write File Track Command

The Write File Track command writes the contents of a variable to one track of a file. The command can either specify that an existing track of an existing file is to be written over or that a new track is to be added onto the end of the file. New tracks are always added to the end of the file. You cannot insert tracks in the middle or at the beginning.

If a new track is being written:

- then the file need not already exist; if necessary, it will be created.
- if the variable is empty (contains the null string) then no new track will be added. However, if the file does not exist, then an empty file will be created.
- if a damaged track is chosen for the new track, then it will be added to the pool of Bad Tracks. A different track will then be selected from the Free tracks Pool.

When rewriting an existing track:

- the first track of the file is "logical" number one. Subsequent tracks are numbered up from this. An error will be reported if you try to write before the first track or after the last one.
- if the variable is empty (contains the null string) then the specified track will be deleted from the file.
- the FDC does not in fact write over the old track. It first writes the new value on a track chosen from the Free Track Pool (rejecting any Bad tracks of course). Only when it has written the track, does it alter the file entry in the directory to point at the new value. The original track is returned to the Free Pool. It is therefore possible to get Disc Full error when rewriting an existing track.

4.4.1 Format - Write File

The format of the Write File Track command is as follows:

```
SAVE "d:filename,T:tracknum,V:variable"
SAVE "d:filename,N,V:variable"
```

Where:

- d is the disc containing the file - "A" or "B"
- filename is the name of the file to be written to
- variable is the name of the variable containing the data which will be written
- tracknum specifies which track of the file should be written the N option specifies that a new track should be added to the end of the file or that a new file should be created.

4.4.2 Examples - Write File

```
SAVE "A:DATA-5,T:7,V:1$"
SAVE "T:1,V:V$,B:CASH"
SAVE "A:SCORES,N,V:L$"
```

4.4.3 Errors - Write File

The following error messages may be generated by the Write File Track command:

- G - the disc is full; all the tracks have been used
- H - the disc or file is write protected
- I - the directory of the disc is full
- J - the specified file does not exist
- M - there is an unrecognizable parameter or the file or variable name has been specified incorrectly or the variable does not exist

- P - an illegal combination of parameters has been given
- R - the track number is out of range
- S - there was an error in reading or writing the directory

4.5 The Delete File Command

The Delete File command deletes the selected file from the directory and returns the tracks used by the file to the pool of free tracks.

A Write Protected file may not be deleted

4.5.1 Format - Delete

The format of the Delete File command is as follows:

```
SAVE "d:filename,D"
```

Where:

- d is the disc containing the file - "A" or "B"
- filename is the name of the file to be changed

4.5.2 Examples - Delete

```
SAVE "A:DATA-5,D"
SAVE "D,B:STARTREK"
```

4.5.3 Errors - Delete

The following error messages may be generated by the Delete File command:

- H - the disc or file is write protected
- J - the specified file does not exist
- M - there is an unrecognizable parameter or the file name has been specified incorrectly
- P - an illegal combination of parameters has been given
- S - there was an error in reading or writing the directory

4.6 The Change File Command

The Change File command is used to Write Protect or Un-protect a file. Any attempt to alter or delete a protected file will fail and will produce an error message.

4.6.1 Format - Change

The format of the Change File command is as follows:

```
SAVE "d:filename,WP"
SAVE "d:filename,WE"
```

Where:

- d is the disc containing the file - "A" or "B"
- filename is the name of the file to be changed
- the WP option will Write Protect the file
- the WE option will Write Enable the file

4.6.2 Examples - Change

```
SAVE "A:DATA-5,WP"
SAVE "S,WP,B:STARTREK"
SAVE "B:X-3,WE"
```

4.6.3 Errors - Change

The following error messages may be generated by the Change File command:

- H - the disc is write protected
- J - the specified file does not exist
- M - there is an unrecognizable parameter or the file name has been specified incorrectly
- P - an illegal combination of parameters has been given
- S - there was an error in reading or writing the directory

4.7 The Format Disc Command

The Format Disc command writes a blank directory to the first track of the selected disc. It does not read or alter the data on the rest of the disc. All the tracks are returned to the pool of free tracks - it is assumed that there are no bad tracks on the disc.

The Format Disc command should be used to initialize every new disc. The command is used to specify the type of the drive and disc being used. The command includes information about

- the number of tracks on one side of the disc
- the number of sides of the disc (1 or 2)
- the density of the disc (single or double)

4.7.1 Format - Format

The format of the Format Disc command is as follows:

```
SAVE "d,FMT,T:trknum"
SAVE "d,FMT,T:trknum,DD"
SAVE "d,FMT,T:trknum,DS"
SAVE "d,FMT,T:trknum,DD,DS"
```

Where

- d is the disc to be formatted - "A" or "B"
- trknum is the number of tracks per side
- the DD option specifies that the disc and drive are double density
- the DS option specifies that the disc and drive are double sided

4.7.2 Examples - Format

```
SAVE "FMT,A,T:33"
SAVE "B,FMT,DS,DD,T:70"
```

4.7.3 Errors - Format

The following error messages may be generated by the Format command:

- H - the disc is write protected
- M - there is an unrecognizable parameter
- P - an illegal combination of parameters has been given
- R - the number of tracks is greater than 124
- S - there was an error in writing the directory

4.8 The Get Directory Command

The Get Directory command is used to print the directory of a disc or read the directory into a string variable. The BASIC program may "slice" into the string to deal with the file descriptions directly.

The directory is actually read from the disc into the computer memory. So you can use this command to ensure that the copy of the directory in RAM is the same as that on the disc in the drive. For instance, you should always do a Get Directory command after changing the disc.

Each file named in the directory is specified by one "line" of the string returned to the string variable. The format of each "line" is as follows:

- the name of the file - padded with blanks to 10 characters
- 1 blank
- the size of the file in tracks - 3 characters

either:

- "WP" if the file is Write protected

or:

- 2 blanks if it is Write Enabled
- 1 blank

either:

- "S" if this is a Startup file

or:

- 1 blank if it isn't
- 1 blank

either:

- "P" if the file is a BASIC program

or:

- 1 blank if it is a data file
- 1 blank

The first file description is

```
,FREE      nnn
```

and gives the number of Free tracks on the disc. If the disc is recorded in double density, the description will be

```
,FREE  DD nnn
```

4.8.1 Format - Directory

The format of the Get Directory command is as follows:

```
LOAD "d,DIR,V:variable"  Load Directory to Variable
LOAD "d,DIR"              Load Directory to Screen
```

Where:

- d is the disc whose directory is to be read - "A" or "B"
- variable is the name of the variable into which the directory will be read. If it is omitted, the directory will be printed.
- the DIR option specifies that the directory should be read

4.8.2 Examples - Directory

```
LOAD "A,DIR,V:S$"
```

```
LOAD "V:A$,DIR,B"
```

4.8.3 Errors - Directory

The following error messages may be generated by the Get Directory command:

- 4 - there is not enough room in memory to create a variable to hold the data
- M - there is an unrecognizable parameter or the variable name has been specified incorrectly
- P - an illegal combination of parameters has been given
- S - there was an error in reading or writing the directory

4.9 The Read Physical Track Command

The Read Physical Track reads the data of one track of the disc into a variable. This command should not be used for normal file input but only if you are maintaining your own disc directory.

An error will be reported if you try to access a track number beyond the last track on the disc.

For side 1 (as opposed to side 0) of double sided discs, you should add 128 onto the track number.

All tracks, including track zero, are read in the density you specified when the disc was Formatted. You will not be able to read a normal directory with a Read Physical Command.

4.9.1 Format - Read Physical

The format of the Read Physical Track command is as follows:

```
LOAD "d,T:tracknum,V:variable"
```

Where:

- d is the disc to be read - "A" or "B"
- variable is the name of the variable into which the track will be read
- tracknum specifies which physical track of the disc should be read

4.9.2 Examples - Read Physical

```
LOAD "A,T:30,V:S$"
LOAD "V:A$,T:0,B"
```

4.9.3 Errors - Read Physical

The following error messages may be generated by the Read Physical Track command:

- 4 - there is not enough room in memory to create a variable to hold the data
- L - there was a serious, non-recoverable, error during reading or writing the track
- M - there is an unrecognizable parameter or the variable name has been specified incorrectly
- R - the track number is out of range

4.10 The Write Physical Track Command

The Write Physical Track writes the contents of a variable to one track of the disc. This command should not be used the last track on the disc.

For side 1 (as opposed to side 0) of double sided discs, you should add 128 onto the track number.

All tracks, including track zero, are written in the density you specified when the disc was Formatted. You will not be able to Write a normal directory with a Write Physical Command.

4.10.1 Format - Write Physical

The format of the Write Physical Track command is as follows:

```
SAVE "d,T:tracknum,V:variable"
```

Where:

- d is the disc to be written to - "A" or "B"
- variable is the name of the variable into which the track will be written
- tracknum specifies which physical track of the disc should be written

4.10.2 Examples - Write Physical

```
SAVE "A,T:30,V:S$"
```

```
SAVE "V:A$,T:0,B"
```

4.10.3 Errors - Write Physical

The following error messages may be generated by the Write Physical Track command:

- H - the disc is write protected
- L - there was a serious, non-recoverable, error during reading or writing the track

- M - there is an unrecognizable parameter or the variable name has been specified incorrectly or the variable does not exist
- P - an illegal combination of parameters has been given
- R - the track number is out of range
- S - there is not enough space in the computer memory to read data into a variable or the data length of a variable to be written is greater than the track length

5.0 Error Reports - All Commands

The FDC software can generate extra error reports other than those listed in the computer BASIC programming handbook. These errors are listed below as well as with the commands which may generate them.

- G - the disc is full; all the tracks have been used
- H - the disc or file is write protected
- I - the directory of the disc is full
- J - the specified file does not exist
- K - the program is too big to be saved; there is no memory left for buffers
- L - there was a serious, non-recoverable, error during reading or writing the track
- M - there is an unrecognizable parameter or the file or variable name has been specified incorrectly or the variable does not exist
- N - the file already exists; the program has not been saved
- P - an illegal combination of parameters has been given
- R - the track number is out of range
- S - there was an error in reading or writing the directory

6.0 Machine Code

If you want to write machine code programs in the area above RAMTOP, you will need to know how the FDC organizes the computer memory.

On reset, the FDC finds the upper limit of the computer memory (in the same way that the computer monitor does) and reserves some workspace immediately below this. It sets RAMTOP to point at the start of the workspace area so BASIC programs can't overwrite it. The first two bytes of the space at RAMTOP point at the actual location of the space used by the FDC. After a reset, this work space starts at RAMTOP + 2.

If you want to put machine code above RAMTOP, you can make space either below the FDC workspace or above it. In either case, you will decrement RAMTOP and you must make a new pointer at RAMTOP to point at the FDC workspace. If the FDC workspace has been moved, you should read the directories by performing a Get Directory.

For example, let's create 100 bytes of space below the FDC workspace. First, find the current value of RAMTOP:

```
LET RAMTOP = PEEK 16388 + 256 * PEEK 16389
```

Now get the pointer to the FDC workspace:

```
LET POINTER = PEEK RAMTOP + 256 * PEEK (RAMTOP + 1)
```

Put in the new value of ramtop:

```
LET RAMTOP = RAMTOP - 100
LET HIGH = INT(POINTER / 256)
LET LOW = RAMTOP - HIGH * 256
POKE 16388, LOW
POKE 16389, HIGH
```

And put back the pointer value:

```
LET HIGH = INT(POINTER / 256)
LET LOW = POINTER - HIGH * 256
POKE RAMTOP, LOW
POKE RAMTOP + 1, HIGH
```

Alternatively, we could move the FDC workspace down by 100 bytes by setting the new pointer to point at RAMTOP + 2. But this time, we must read the directories into the new workspace:

```
LET RAMTOP = PEEK 16388 + 256 * PEEK 16389
LET RAMTOP = RAMTOP - 100
LET HIGH = INT(RAMTOP / 256)
LET LOW = RAMTOP - HIGH * 256
POKE 16388, LOW
POKE 16389, HIGH
```

```
LET POINTER = RAMTOP + 2
LET HIGH = INT(POINTER / 256)
LET LOW = POINTER - HIGH * 256
POKE RAMTOP, LOW
POKE RAMTOP + 1, HIGH
LOAD "A,DIR,V:S$"
LOAD "B,DIR,V:S$"
```

Here is a diagram of how the computer memory is set up:

